Objective C Programming

Lecture 1
Introduction. Xcode.

International University of Information Technology
Almaty, 2012

Hello!

My name is Rakhim Davletkaliyev, BSc

r.davletkaliyev@iitu.kz

http://freetonik.com

About the course

- Objective C, Foundation, Cocoa
- Building apps for Mac OS X
- Pre- and post requisites

- http://wikistan.ru/blog/macosdev/

Course structure

Lectures: YouTube and iTunes

Labs: 15 labs, one point each – 15 points
Assignments: 4 projects, 5 points each – 20 points
Written quizzes: 4 quizzes, 3.75 points each – 15 points

Midterm – 10 points
Final – 40 points

Office hours
Policy regarding late submission and plagiarism.

What is Objective C?

- Object-oriented language with Smalltalk-style messaging system on top of C
- Used by Apple in OS X and iOS

Hello, world!


Color palette
Transparency
Effects
Measuring tools
• Text spacing
• Alignment
Writing
Tools
PROJECT: APPLICATION.APP
blog.tice.de
ARCHITECT: XCODE
#1

type id

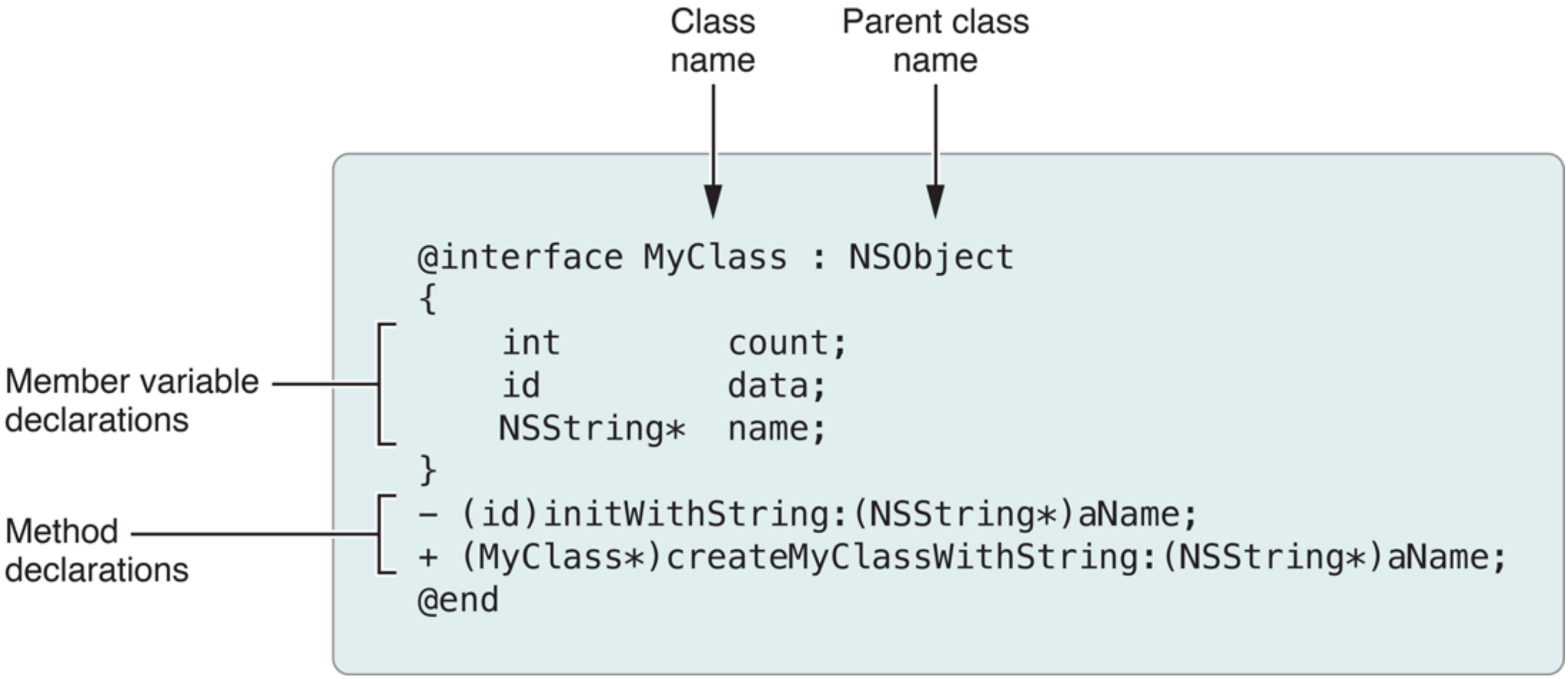
- id data type is used to store an object of any type

```
int height;
float temperature;
id graphicObject;
```

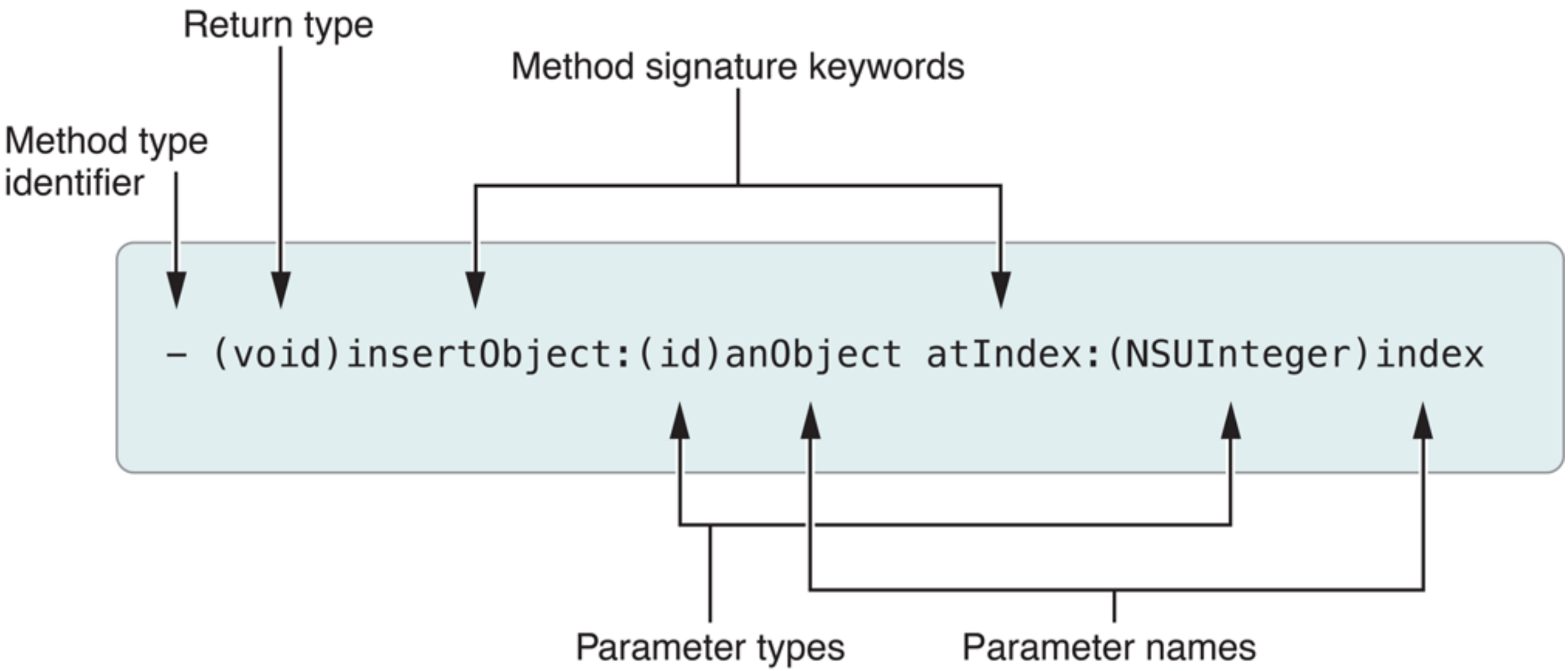
Files

- .h – header files
- .m – source files
- .mn – source files if C++ is present

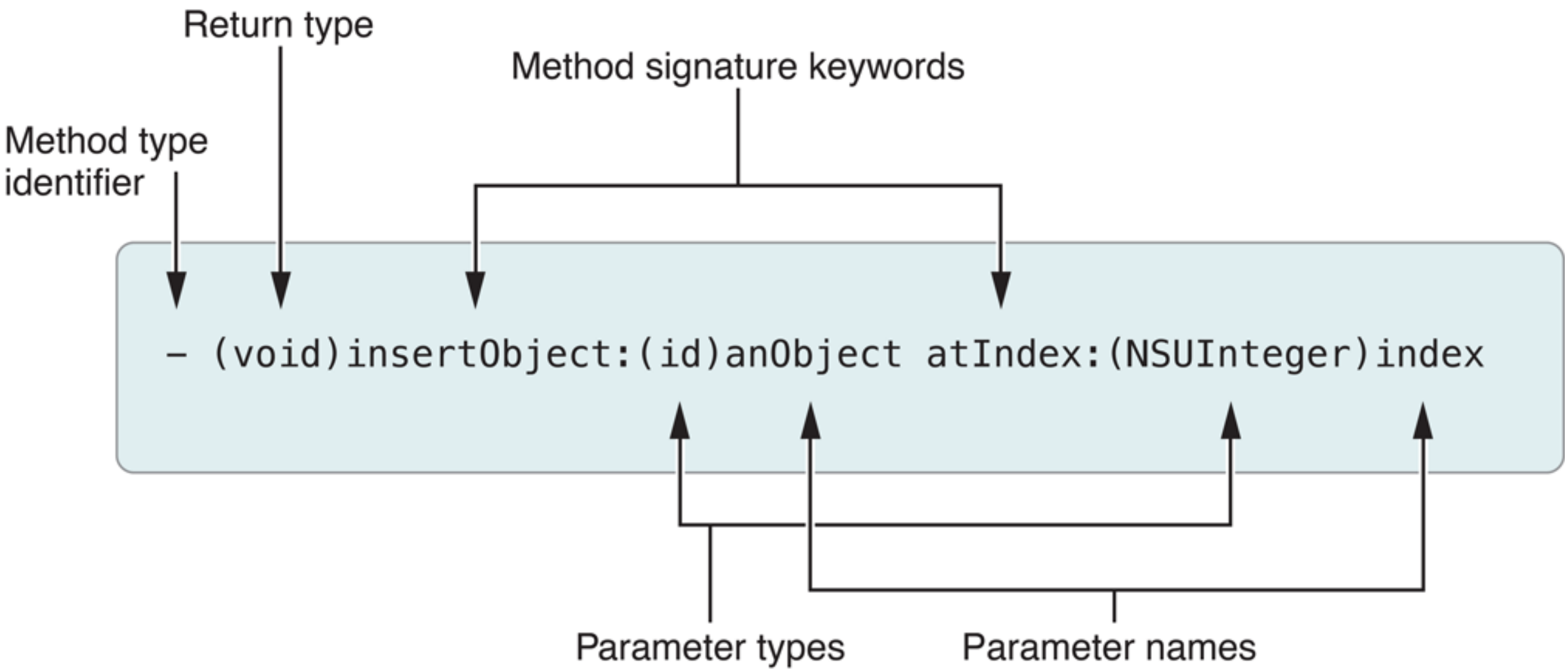
Class declaration



Methods


Return type
Method signature keywords
Method type identifier
- (void)insertObject:(id)anObject atIndex:(NSUInteger)index
Parameter types
Parameter names

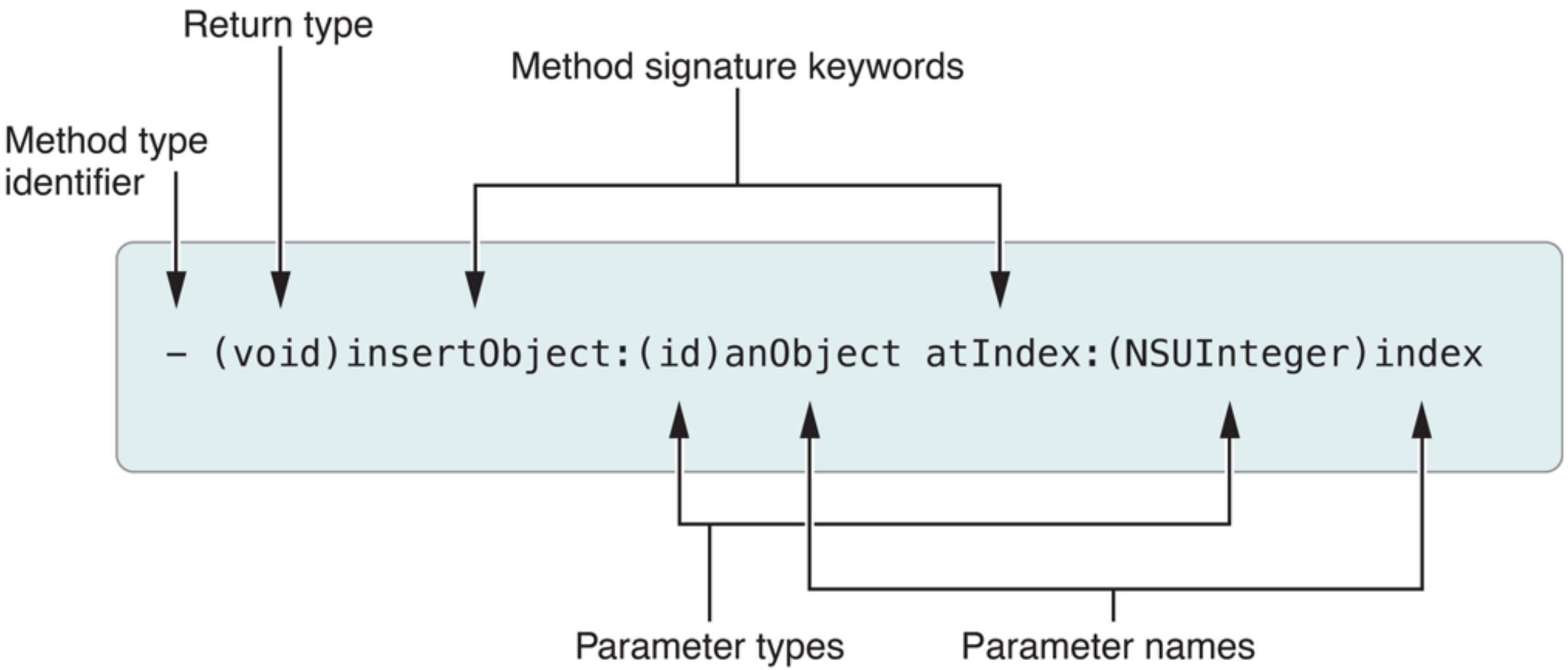
Methods



vs

void insertObjectAtIndex (object, index);

Methods



method's name is

insertObject:atIndex:

Methods

[Class method];

[Instance method];

[receiver message];

myCar = [Car new];

[myCar start];

example

```
[myArray insertObject:anObject atIndex:0];
```

example

```
[myArray insertObject:myObject atIndex:0];
```

nesting

```
[[myAppObject theArray] insertObject:[myAppObject objectToInsert] atIndex:0];

[anArray insertObject:[myAppObject objectToInsert] atIndex:0];

[anArray insertObject:anObject atIndex:0];
```

dot notation

```
[myAppObject.theArray insertObject:myAppObject.objectToInsert atIndex:0];
```

Objective C vs C++

```
@interface Human : Animal {
 // instance variables
 int age;
}
+ classMethod1;
+ (return_type)classMethod2;
+ (return_type)classMethod3:(param1_type)param1_varName;

- (return_type)instanceMethod1:(param1_type)param1_varName :(param2_type)param2_varName;
- (return_type)instanceMethod2WithParameter :(param1_type)param1_varName otherParameter:
(param2_type)param2_varName;
@end
```

```
class Human : public Animal {
 protected:
 // instance variables
 int age;

 public:
 // Class (static) functions
 static void * classMethod1();
 static return_type classMethod2();
 static return_type classMethod3(param1_type param1_varName);

 // Instance (member) functions
 return_type instanceMethod1 (param1_type param1_varName, param2_type param2_varName);
 return_type instanceMethod2WithParameter (param1_type param1_varName, param2_type
param2_varName=default);
};
```

@implementation

```
@implementation NewClassName{
  memberDeclarations;
}

methodDefinitions;
@end
```

FractionsClass example: @interface

```
#import <Foundation/Foundation.h>

@interface Fraction: NSObject

-(void) print;
-(void) setNumerator: (int) n;
-(void) setDenominator: (int) d;

@end
```

FractionsClass example: @implementation

```
@implementation Fraction {
   int numerator;
   int denominator;
}
-(void) print {
   NSLog (@"%i/%i", numerator, denominator);
}

-(void) setNumerator: (int) n {
numerator = n;
}

-(void) setDenominator: (int) d {
denominator = d;
}
@end
```

FractionsClass example: program code

```
int main (int argc, char * argv[]) {
    @autoreleasepool {
        Fraction *myFraction;

        // Create an instance of a Fraction
        myFraction = [Fraction alloc];
        myFraction = [myFraction init];

        // Set fraction to 1/3
        [myFraction setNumerator: 1];
        [myFraction setDenominator: 3];

        // Display the fraction using the print method
        NSLog (@"The value of myFraction is:");
        [myFraction print];
}

return 0;
}
```